Mean field approach to learning in Boltzmann Machines

H.J. Kappen*and F. B. Rodríguez†

August 28, 1997

Abstract

The learning process in Boltzmann Machines is computationally very expensive. The computational complexity of the exact algorithm is exponential in the number of neurons. We present a new approximate learning algorithm for Boltzmann Machines, which is based on mean field theory and the linear response theorem. The computational complexity of the algorithm is cubic in the number of neurons.

In the absence of hidden units, we show how the weights can be directly computed from the fixed point equation of the learning rules. Thus, in this case we do not need to use a gradient descent procedure for the learning process. We show that the solutions of this method are close to the optimal solutions and give a significant improvement over the naive mean field approach. The method is of immediate relevance for learning in probabilistic approaches, such as Bayesian networks.

Keywords: Mean field theory, probability models

1 Introduction

Boltzmann Machines (BMs) (Ackley et al., 1985), are networks of binary neurons with a stochastic neuron dynamics, known as Glauber dynamics. Assuming symmetric connections between neurons, the probability distribution over neuron states $\vec{s}$ will become stationary and will be given by the Boltzmann-Gibbs distribution $P(\vec{s})$. The Boltzmann distribution is a known function of the weights and thresholds of the network. However, computation of $P(\vec{s})$ or any statistics involving $P(\vec{s})$, such as mean firing rates or correlations, requires

*RWCP SNN Laboratory, Department of Biophysics, University of Nijmegen, Geert Grooteplein 21, NL 6525 EZ Nijmegen, The Netherlands

†Instituto de Ingenier´ıa del Conocimiento & Departamento de Ingenier´ıa Informática, Universidad Autónoma de Madrid, Canto Blanco,28049 Madrid, Spain.

exponential time in the number of neurons. This is due to the fact that $P(\vec{s})$ contains a normalization term Z, which involves a sum over all states in the network, of which there are exponentially many. This problem is particularly important for BM learning. This is because the BM learning rule requires the computation of correlations between neurons. Thus, learning in BMs requires exponential time.

A well-known approximate method to compute Z, or any other statistics, is by importance sampling (Itzykson and Drouffe, 1989). Glauber dynamics is an example of importance sampling. Importance sampling is more effective than the exact computation because the sampling is biased towards the parts of the configuration space that will give the dominant contribution to Z, but is still very time consuming. This is the approach chosen for learning in the original Boltzmann Machine (Ackley et al., 1985). The method has poor convergence and can only be applied to small networks.

In (Peterson and Anderson, 1987), an acceleration method for learning in BMs is proposed. They suggest to replace the correlations by the naive mean field approximation: $\langle s_i s_j \rangle = m_i m_j$, where m_i is the mean field activity of neuron i. The mean fields are given by the solution of a set of n coupled mean field equations, with n the number of neurons. The solution can be efficiently obtained by fixed point iteration. The method was further elaborated in (Hinton, 1989).

It can be shown (Kappen and Rodríguez, 1997) that the naive mean field approximation of the learning rules does not converge in general. Furthermore, we argue that in the correct treatment of mean field, the correlations can be computed using the linear response theorem (Parisi, 1988). In the context of neural networks this approach was first introduced by (Ginzburg and Sompolinsky, 1994) for the computation of time-delayed correlations and later by (Kappen, 1997) for the computation of stimulus dependent correlations.

2 Boltzmann Machine learning

The Boltzmann Machine is defined as follows. The possible configurations of the network can be characterized by a vector $\vec{s} = (s_1, .., s_i, .., s_n)$, where $s_i = \pm 1$ is the state of the neuron i, and n the total number of the neurons. Neurons are updated using Glauber dynamics.

Let us define the energy of a configuration $\vec{s}$ as

$$-E(\vec{s}) = \frac{1}{2} \sum_{i,j} w_{ij} s_i s_j + \sum_i s_i \theta_i. \tag{1}$$

w_{ij} and θ_i denote the weights and thresholds in the network.

The probability to find the network in a state $\vec{s}$ converges to a stationary

distribution (thermal equilibrium) and is given by the Boltzmann distribution

$$p(\vec{s}) = \frac{1}{Z}\exp\{-E(\vec{s})\}. \tag{2}$$

$Z = \sum_{\vec{s}} \exp\{-\beta E(\vec{s})\}$ is the partition function which normalizes the probability distribution.

A learning rule for Boltzmann Machines was introduced by Ackley, Hinton and Sejnowski (Ackley et al., 1985). Let us partition the neurons in a set of n_v visible units and n_h hidden units ($n_v + n_h = n$). Let α and β label the 2^{n_v} visible and 2^{n_h} hidden states of the network, respectively. Thus, every state $\vec{s}$ is uniquely described by a tuple $\alpha\beta$. Learning consists of adjusting the weighs and thresholds in such a way that the Boltzmann distribution on the visible units $p_\alpha = \sum_\beta p_{\alpha\beta}$ approximates a target distribution q_α as closely as possible.

A suitable measure of the difference between the distributions p_α and q_α is the Kullback divergence (Kullback, 1959)

$$K = \sum_\alpha q_\alpha \log \frac{q_\alpha}{p_\alpha}. \tag{3}$$

It is easy to show that $K \geq 0$ for all distributions p_α and $K = 0$ iff $p_\alpha = q_\alpha$ for all α.

Therefore, learning consists of minimizing K using gradient descent, and the learning rules are given by (Ackley et al., 1985; Hertz et al., 1991)

$$\Delta w_{ij} = \eta\Big(\langle s_i s_j \rangle_c - \langle s_i s_j \rangle \Big), \quad \Delta\theta_i = \eta\Big(\langle s_i \rangle_c - \langle s_i \rangle \Big). \tag{4}$$

The parameter η is the learning rate. The brackets $\langle\cdot\rangle$ and $\langle\cdot\rangle_c$ denote the 'free' and 'clamped' expectation values, respectively. The 'free' expectation values are defined as usual:

$$\begin{aligned} \langle s_i \rangle &= \sum_{\alpha\beta} s_i^{\alpha\beta} p_{\alpha\beta} \\ \langle s_i s_j \rangle &= \sum_{\vec{s}} s_i^{\alpha\beta} s_j^{\alpha\beta} p_{\alpha\beta}. \end{aligned} \tag{5}$$

The 'clamped' expectation values are obtained by clamping the visible units in a state α and taking the expectation value with respect to q_α:

$$\begin{aligned} \langle s_i \rangle_c &= \sum_{\alpha\beta} s_i^{\alpha\beta} q_\alpha p_{\beta|\alpha} \\ \langle s_i s_j \rangle_c &= \sum_{\alpha\beta} s_i^{\alpha\beta} s_j^{\alpha\beta} q_\alpha p_{\beta|\alpha} \end{aligned} \tag{6}$$

$s_i^{\alpha\beta}$ is the value of neuron i when the network is in state $\alpha\beta$. $p_{\beta|\alpha}$ is the conditional probability to observe hidden state β given that the visible state is α. Note that in Eqs. 4–6, i and j run over both visible and hidden units.

Thus, the BM learning rules contain clamped and free expectation values of the Boltzmann distribution. The computation of the free expectation values is intractible, because the sums in Eqs. 5 consist of 2^n terms. If q_α is given in the form of a training set of p patterns, the computation of the clamped expectation values, Eqs. 6, contains $p2^{n_h}$ terms. This is intractible as well, but usually less expensive than the free expectation values. As a result, the BM learning algorithm cannot be applied to practical problems.

3 The mean field method and the linear response correction

The basic idea of mean field theory is to replace the quadratic term in the energy, $w_{ij}s_is_j$ in Eq. 1, by a term linear in s_i. Such a linearized form allows for efficient computation of the sum over all states, such as Eqs. 5 and 6 and the partition function Z. We define the mean field energy

$$-E_{mf}(\vec{s}) = \sum_i s_i\{W_i + \theta_i\} \tag{7}$$

where we introduce n mean fields W_i. The mean fields approximate the lateral interaction between neurons. The values of W_i must be chosen such that this approximation is as good as possible. Following the standard mean field approach (Itzykson and Drouffe, 1989) the approximate free energy is given as

$$-F = \log Z' = \sum_i \log(2\cosh(\theta_i + W_i)) - \sum_i W_i m_i + \frac{1}{2}\sum_{i,j} w_{ij}m_im_j \tag{8}$$

with $m_i = \tanh(W_i + \theta_i)$. The mean fields are given by minimizing the free energy which gives the coupled set of mean field equations:

$$m_i = \tanh(\sum_j w_{ij}m_j + \theta_i) \tag{9}$$

We can now compute the mean firing rates and correlations in the mean field approximation:

$$\langle s_i\rangle = \frac{1}{Z}\frac{dZ}{d\theta_j} \approx \frac{1}{Z'}\frac{dZ'}{d\theta_j}, \quad \langle s_is_j\rangle \approx \frac{1}{Z'}\frac{d^2Z'}{d\theta_id\theta_j} \tag{10}$$

While computing $\frac{dZ}{d\theta_j}$, using Eq. 8, we must be aware that the mean fields W_i depend on θ_i through Eq. 9:

$$\langle s_i\rangle \approx \frac{d}{d\theta_i}\log Z' = \left(\frac{\partial}{\partial\theta_i} + \sum_j \frac{\partial W_j}{\partial\theta_i}\frac{\partial}{\partial W_j}\right)\log Z' = m_i \tag{11}$$

$$\langle s_is_j\rangle \approx \frac{1}{Z'}\frac{d}{d\theta_j}(Z'm_i) = m_im_j + A_{ij} \tag{12}$$

with $A_{ij} = \frac{dm_i}{d\theta_j}$. The last step in Eq. 11 follows when we use the mean field equations Eq. 9. Eq. 12 is known as the linear response theorem (Parisi, 1988). The inverse of the matrix A can be directly obtained by differentiating Eq. 9 with respect to θ_i. The result is:

$$(A^{-1})_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - w_{ij} \tag{13}$$

Thus, our approximation consists of replacing the free expectation values in Eqs. 4 by their linear response approximations Eqs. 9, 11-13. The clamped quantities are directly computed from the data. The inclusion of hidden units is straigthforward and is discussed elsewhere (Kappen and Rodríguez, 1997). The complexity of the method is dominated by the computations in the free phase. The computation of the linear response correlations involves the inversion of the matrix A, which requires $\mathcal{O}(n^3)$ operations. The computation of the mean firing rates through fixed point iteration of Eq. 9 requires $\mathcal{O}(n^2)$ or $\mathcal{O}(n^2 \log n)$ operations, depending on whether fixed precision in the components of m_i or in the vector norm $\sum_i m_i^2$ is required. Thus, the full mean field approximation, including the linear response correction, computes the gradients in $\mathcal{O}(n^3)$ operations.

3.1 No hidden units

For the special case of a network without hidden units we can make significant simplifications. In this case, the gradients Eqs. 4 can be set equal to zero and can be solved directly in terms of the weights and thresholds, i.e. no 'gradient based learning' is required. First note that $\langle s_i \rangle_c$ and $\langle s_i s_j \rangle_c$ can be computed exactly from the data for all i and j. Let us define $C_{ij} = \langle s_i s_j \rangle_c - \langle s_i \rangle_c \langle s_j \rangle_c$.

The fixed point equation for $\Delta\theta_i$ gives

$$\Delta\theta_i = 0 \Leftrightarrow m_i = \langle s_i \rangle_c . \tag{14}$$

The fixed point equation for Δw_{ij}, using Eq. 14, gives

$$\Delta w_{ij} = 0 \Leftrightarrow A_{ij} = C_{ij}, i \neq j. \tag{15}$$

The fixed point equations are only imposed for the off-diagonal elements of Δw_{ij} because the Boltzmann distribution Eq. 2 does not depend on the diagonal elements w_{ii}. The condition $\Delta w_{ii} = 0$ is automatically satified in the exact method. However, in the approximate method things are different. The solution depends on w_{ii} in Eq. 9 and the condition $\Delta w_{ii} = 0$ must be enforced explicitly to ensure that $1 = \langle s_i^2 \rangle = 1 - m_i^2 - A_{ii}$. Thus, instead of Eq. 15, one must impose the stronger condition $A_{ij} = C_{ij}$ for all i, j, which is equivalent to $(A^{-1})_{ij} = (C^{-1})_{ij}$. Using Eq. 13 we obtain

$$w_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - (C^{-1})_{ij} \tag{16}$$

In this way we have solved m_i and w_{ij} directly from the fixed point equations. The thresholds θ_i can now be computed from Eq. 9:

$$\theta_i = \tanh^{-1}(m_i) - \sum_j w_{ij} m_j \tag{17}$$

Note that this method does not require fixed point iterations to obtain mean firing rates m_i in terms of w_{ij} and θ_i. Instead, the 'inverse' computation of θ_i given m_i and w_{ij} is required in Eq. 17.

4 Results

In this Section we will compare the accuracy of the linear response correction with the exact method and with the naive mean field approximation. We restrict ourselves to networks without hidden units. Of course, there are many probability estimation problems, for which the BM without hidden units is a poor model. The optimal solution can be found using the exact gradient descent method. Our main concern is whether the linear response approximation will give a solution which is sufficiently close to the optimal solution, and not whether the optimal solution is good or bad.

The correct way to compare our method to the exact method is by means of the Kullback divergence. However, this comparison can only be done for small networks. The reason is that the computation of the Kullback divergence requires the computation of the Boltzmann distribution, Eq. 2, which requires exponential time due to the partition function Z. In addition, the exact learning method requires exponential time. The comparison by Kullback divergence on small problems is the subject of Section 4.1.

For networks with a large number of units one can demonstrate the quality of the linear response method by means of a pattern completion task i.e. the network must be able to generate the rest of a pattern, when part of the pattern is shown. The results on this pattern completion task suggest that the performance of the linear response method is also good for large networks (Kappen and Rodríguez, 1997).

4.1 Comparison using Kullback divergence

In order to show the performance of the linear response correction, we have compared it with the results obtained with the exact method and with a 'mean field' method that ignores correlations. For the exact method (K_{ex}), we have used a gradient descent method with a momentum term. The mean firing rates and correlations are computed using Eqs. 5. For the linear response method (K_{lr}), we obtain the weights and thresholds from Eq. 16 and Eq. 17. In the case

of the naive mean field approximation (K_{mf}), we assume a factorized model:

$$P_{mf}(\vec{s}) = \Pi_i \frac{1}{2}(1 + s_i m_i). \tag{18}$$

The mean firing rates are given by $m_i = \langle s_i \rangle_c$.

We compared the methods on a number of typical examples in Fig. 1a. Each neuron value $s_i^\mu = \pm 1, i = 1, \ldots, n, \mu = 1, \ldots, p$ is generated randomly and independently with equal probability. The three methods are compared by computing the Kullback divergence, using Eq. 3, that we obtain for each method on each of the data sets. The network size was varied from 3 to 10 neurons. For each data set we compute $K_{lr} - K_{ex}$ and $K_{mf} - K_{ex}$. In the Figure, we show these values averaged over all data sets, as well as their variances.

The difference in quality between the exact method and the linear response method is a sensitive function of the number of patterns in the data set. This is illustrated for $n = 6$ in Fig.1b.

We conclude that the linear response correction gives a good approximation of the exact results. The naive mean field approximation that ignores the correlations is much worse, as should be expected. It indicates that correlations play a significant role in these learning problems.

5 Discussion

We have proposed a new efficient method for learning in Boltzmann Machines. The method is generally applicable to networks with or without hidden units. It makes use of the linear response theorem for the computation of the correlations within the mean field framework. In our view, this is the proper way to compute correlations in the mean field framework, instead of the 'naive' mean field assumption $\langle s_i s_j \rangle = \langle s_i \rangle \langle s_j \rangle$ which has been advocated by some authors (Peterson and Anderson, 1987; Hinton, 1989; Hinton et al., 1995; Dayan et al., 1995).

We have derived an explicit expression for the optimal weights and thresholds for networks without hidden units. Thus, no gradient descent procedure is needed. In our numerical results we have restricted ourselves to networks without hidden units. We argue that this is sufficient to show the advantage of the method, since the 'free' expectation values are the most time consuming part of the computation. These expectation values are unaffected by the fact whether part of the network is hidden or visible.

In the presence of hidden units, both the exact method and the linear response method require a gradient descent algorithm. The advantage of our method is that the gradients can be computed in $\mathcal{O}(n^3)$, instead of in $\mathcal{O}(2^n)$, time. The required number of iterations may be somewhat more for the linear response method, because the gradients are only computed approximately.

This brings us to an interesting point, which is the convergence of the gradient descent algorithm in the linear response approximation. Convergence requires the existence of a Lyapunov function. The Kullback divergence is clearly a Lyapunov function for the exact method, but we were not able to find a Lyapunov function for the linear response approximation. In fact, one would like to construct a cost function such that its gradients are equal to the gradients of K in the linear response approximation. Whether such a function exists is unknown to our knowledge.

An important potential application domain is for Bayesian networks (Pearl, 1988). These networks encode domain knowledge in a graphical structure. It is well known, that inference and learning in Bayes networks is intractible (Cooper, 1990). The most efficient algorithms transform the directed graph by a number of steps to an undirected graph (Lauritzen and Spiegelhalter, 1988). The remaining complexity is in the estimation of the joint probability distribution on cliques in the undirected graph. The Boltzmann Machines as used in this paper could be used to estimate the probability distributions on these cliques. This would result in a polynomial time learning algorithm for Bayesian networks.

Acknowledgement

We would like to thank Martijn Leisink for producing Fig. 1b.

References

Ackley, D., Hinton, G., and Sejnowski, T. (1985). A learning algorithm for Boltzmann Machines. *Cognitive Science*, 9:147–169.

Cooper, G. (1990). The computational complexity of probabilistic inferences. *Artificial Intelligence*, 42:393–405.

Dayan, P., Hinton, G., Neal, R., and Zemel, R. (1995). The Helmholtz Machine. *Neural Computation*, 7:889–904.

Ginzburg, I. and Sompolinsky, H. (1994). Theory of correlations in stochastic neural networks. *Physical Review E*, 50:3171–3191.

Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the theory of neural computation*, volume 1 of *Santa Fe Institute*. Addison-Wesley, Redwood City.

Hinton, G. (1989). Deterministic Boltzmann learning performs steepest descent in weight-space. *Neural Computation*, 1:143–150.

Hinton, G., Dayan, P., Frey, B., and Neal, R. (1995). The "Wake-Sleep" Algorithm for Unsupervised Neural Networks. *Science*, 268:1158–1161.

Itzykson, C. and Drouffe, J.-M. (1989). *Statistical Field Theory*. Cambridge monographs on mathematical physics. Cambridge University Press, Cambridge, UK.

Kappen, H. (1997). Stimulus dependent correlations in stochastic networks. *Physical Review E*, 55:5849–5858.

Kappen, H. and Rodríguez, F. (1997). Efficient learning in Boltzmann Machines using linear response theory. *Neural Computation*. Submitted.

Kullback, S. (1959). *Information Theory and Statistics*. Wiley, New York.

Lauritzen, S. and Spiegelhalter, D. (1988). Local computations with probabilties on graphical structures and their application to expert systems. *J. Royal Statistical society B*, 50:154–227.

Parisi, G. (1988). *Statistical Field Theory*. Frontiers in Physics. Addison-Wesley.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, California.

Peterson, C. and Anderson, J. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019.

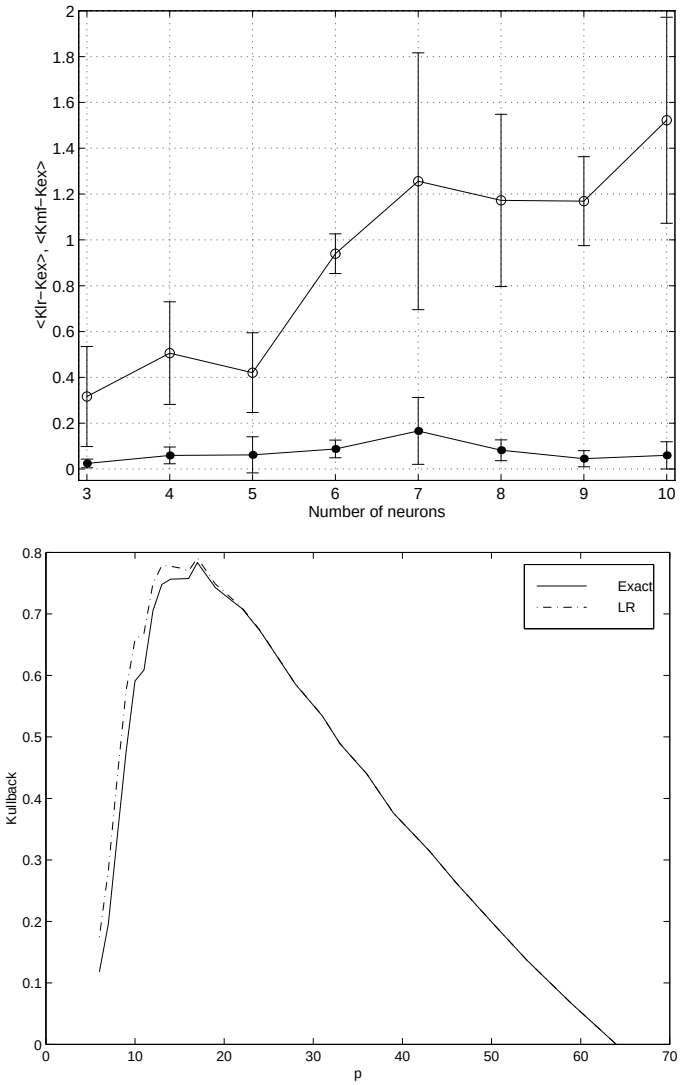

Figure 1: a) Kullback divergence relative to exact method, for mean field approximation (open symbols) and linear response method (black symbols). The number of patterns $p = 2n$. Results are averaged over 4 data sets. The error bars indicate the variance over the data sets. b) Kullback divergence for the exact method and the linear response method for $n = 6$ as a function of the number of patterns in the trainingset. The results are averaged over 100 runs. The errorbars in the difference $K_{ex} - K_{lr}$ are of order 0.05